

■ ■ ■ Troubleshooting Oracle Performance Using Embarcadero DB Optimizer XE



Udo Fohrmann
udo.fohrmann@trivadis.com

Philipp Salvisberg
philipp.salvisberg@trivadis.com

TechEvent
Regensdorf, 15th April 2011

trivadis
makes IT easier. ■ ■ ■

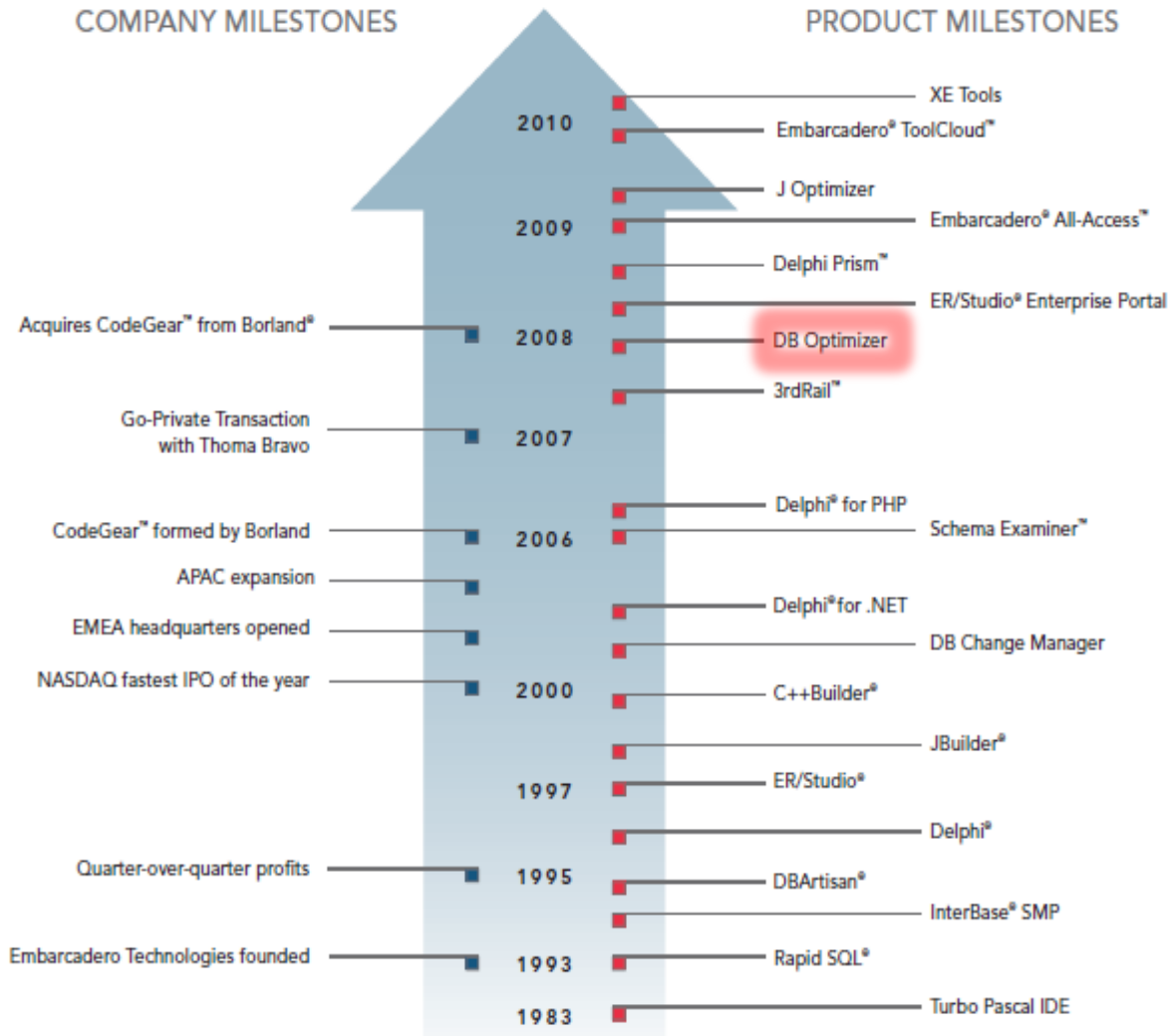
Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

Embarcadero Technologies



Kyle Hailey on <http://dboptimizer.com>:

"Most recently I worked at Embarcadero Technologies where I designed the first successful product the company has written internally and released in the last 10 years called DB Optimizer.

DB Optimizer was inspired by the successes and frustrations I had working on Oracle's EM 10g. In OEM 10g I worked on a complete redesign of the Oracle Enterprise Manager performance pages, shifting the screens away from confusing clutter to simple but powerful graphics based on wait time and a new metric session load (aka AAS – average active sessions). The successful redesign of OEM 10g has continued to be the foundation of OEM 11g"

After 5 ½ years Kyle left Embarcadero Technologies in October 2010 and works now as Performance Architect at Delphix. Before his stage at Embarcadero he worked in the Oracle Enterprise Manager 10g team along with John Beresniewicz, Graham Wood and Gaja Vaidyanatha.

DB Optimizer XE Product Overview

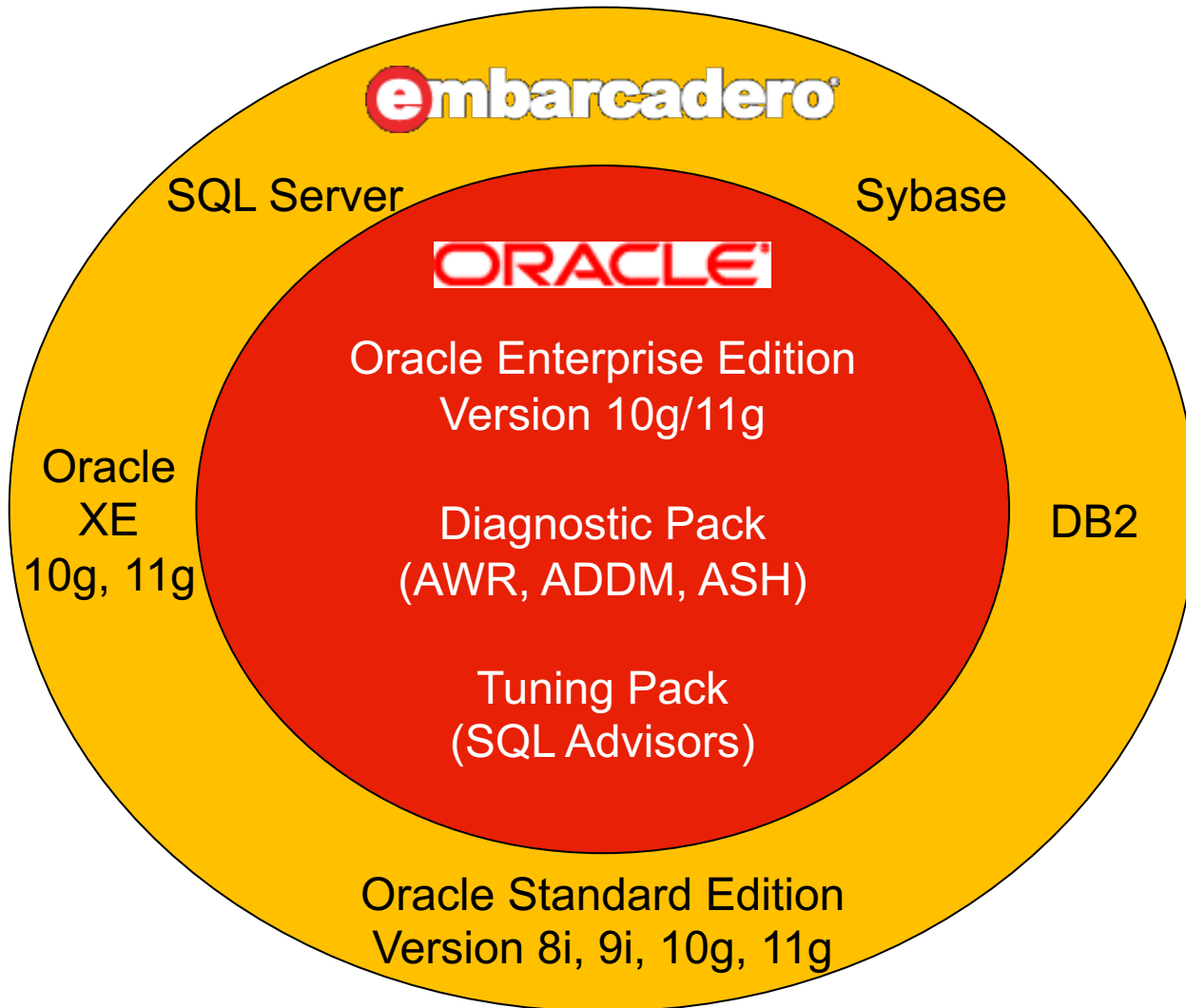


- Major release 2010

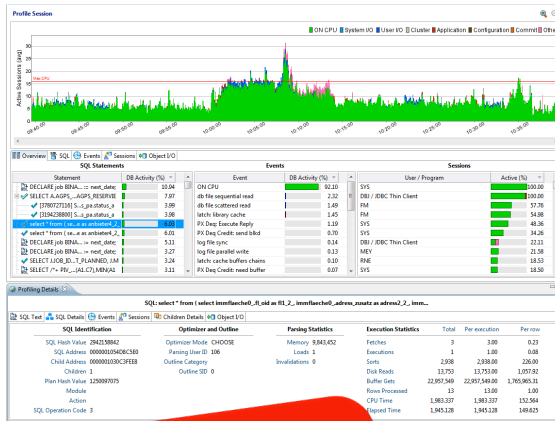
	DB Optimizer XE Developer	DB Optimizer XE Pro	DB Optimizer XE Enterprise	DB Optimizer Professional
Support for all database platforms included	X	X	X	Optional
Embarcadero ToolCloud	X	X	X	Optional
InstantOn™	X	X	X	Optional
Centralized license management	X	X	X	Optional
DB Optimizer 2.X core	X	X	X	X
SQL Profiling	X	X	X	X
Load Testing	X	X	X	X
SQL Editor	X	X	X	X
Tuning: -Visual SQL Tuning -SQL Hint Analysis	-	X	X	X
Profiling into Repository (single data source)	-	X	X	X
24x7 Monitoring Server, DB Performance Center XE Client*	-	X	X	-
24x7 Monitoring Server, DB Performance Center XE Server	-	-	X	-

*The DB Performance Center XE Client requires at least one DB Performance Center XE Server, which is part of DB Optimizer XE Enterprise

Why, When to Use DB Optimizer XE?



Approach



Overview

Generate cases Perform detail analysis Execute each generated case 5 times

Statement	Time	Analysis
Name	Schema	Text
SELECT 1	SH	select from > joined

Generated Cases									
SQL Statements and Cases		Cost	Execution Statistics			Other Execution Statistics			
Name	Text	Value	Elapsed Time (s)	% Result	Rows Returned	Physical Reads	Logical Reads	CPU Time (s)	
SELECT 1	select from > joined tables, sales	213.0	1.40	100%	18	646	998	0.38	
LEADING1		203.0	2.04	100%	18	8740	998	0.25	
LEADING2		3084.0	1.77	100%	18	8710	998	0.17	
INDEX_FFS		213.0	1.40	100%	18	7772	998	0.15	
LEADING4		2178.0	1.58	100%	18	7912	998	0.15	
FULL		213.0	1.56	100%	18	7424	998	0.15	
INDEX		78627.0	1.56	100%	18	2364	5072	0.08	
USE_AL		161580.0	1.40	100%	18	7	40373	0.13	
ORHED		14007.0	0.86	100%	18	337	18862	0.06	
LEADING5		12840.0	0.77	100%	18	702	18852	0.06	
INDEX_COMBINE	click to view SQL text	22571.0	0.45	100%	18	7	18024	0.06	

```

sh@phs11 Change data source
Ad Hoc SQL   SQL file

SELECT
  /*+ INDEX_COMBINE (c CO COUNTRIES_PK ) INDEX_COMBIN
  co.country_region,
  t.fiscal_year,
  COUNT (*) AS number_of_sales,
  COUNT (DISTINCT p.prod_id) AS distinct_products,
  COUNT (DISTINCT c.cust_id) AS distinct_customers
FROM
  sales s
  INNER JOIN
  customers c
  ON c.cust_id = s.cust_id
  INNER JOIN
  countries co
  ON co.country_id = c.country_id
  INNER JOIN
  products p
  ON p.prod_id = s.prod_id
  INNER JOIN
  times t
  ON t.time_id = s.time_id
WHERE
  co.country_region IN ('Oceania', 'Asia') AND
  p.prod_category_desc = 'Phoco'
GROUP BY
  co.country_region,
  t.fiscal_year
ORDER BY
  co.country_region,
  t.fiscal_year
  
```

Number of parallel sessions: 15

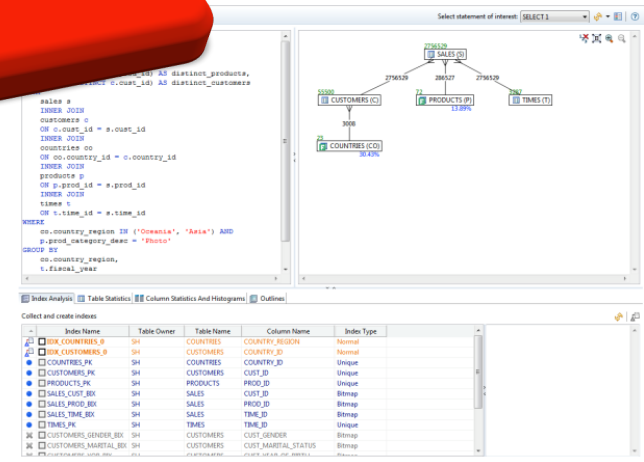
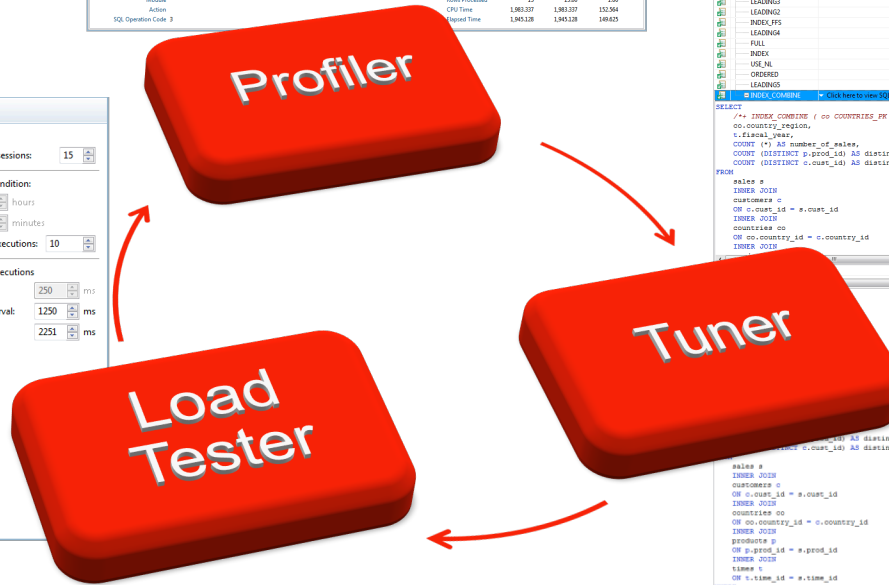
Execution end condition:

- Time: 0 hours
- 10 minutes

Number of executions: 10

Sleep between executions

- Fixed delay: 250 ms
- Random interval: 1250 ms
- 2251 ms



Functional Overview



■ Profiler

- Visualize resource usage per average active session, SQL statements, events, sessions, object I/O
- Life data capturing (fixed sampling interval of 1 second, configurable refresh interval)
- Save captured data for offline analysis to flat file or database

■ Tuner

- Paste in SQL to be tuned or extract them from scripts, DB objects or SGA
- Find faster plans using hints
- Apply faster plans (stored outlines)
- Index, statistics, outline analysis
- Index advisor
- Visual SQL Tuning (VST diagrams)
- Explain plain incl. binds/user

■ Load Tester

- Run a chosen SQL in parallel
 - Define number of parallel sessions
 - Define number of executions or run time
 - Define execution delay
- Monitor execution with Profiler
- Compare original SQL with tuned variants within Profiler (time based identification)

■ SQL Editor

- Code assist (e.g. code completion)
- Code formatter
- Code corrections and transformations (quick fixes)

Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

Installation



- Packaged into a single-file (Eclipse Rich Client Application)
 - ▣ Embarcadero DB Optimizer XE for Windows (used for this evaluation)
 - ▣ Embarcadero DB Optimizer XE for Linux
 - ▣ Embarcadero DB Optimizer XE InstantOn for All-Access for Windows
- Runs without installation
- Argue that they do not change the OS-settings/registry (which is in fact not true)
- Licensing
 - ▣ Default: Named User bound on a Workstation (multiple installation on desktop and notebook are possible, but have to be approved by Embarcadero Technologies)
 - ▣ Other, more feasible network based licensing models available

Privileges Needed on Target Database Instance

- ■ ■
- For Profiling only
 - CREATE SESSION
 - SELECT_CATALOG_ROLE
- For Explain Plan
 - CREATE TABLE (to create an explain plan table)
 - Additional privileged depending on statement to be explained
- SQLTUNING role is recommended for full functionality (see next slide)

```
CREATE USER TUNINGUSER IDENTIFIED BY tuningpassword
DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP
QUOTA UNLIMITED ON USERS
PROFILE DEFAULT
ACCOUNT UNLOCK;
```

```
GRANT SQLTUNING TO TUNINGUSER;
ALTER USER TUNINGUSER DEFAULT ROLE SQLTUNING;
```

SQLTUNING ROLE – Privileges



```
CREATE ROLE SQLTUNING NOT IDENTIFIED;  
GRANT SQLTUNING TO CONNECT;  
GRANT SQLTUNING TO SELECT_CATALOG_ROLE;  
GRANT ANALYZE ANY TO SQLTUNING;  
GRANT CREATE ANY OUTLINE TO SQLTUNING;  
GRANT CREATE ANY PROCEDURE TO SQLTUNING;  
GRANT CREATE ANY TABLE TO SQLTUNING;  
GRANT CREATE ANY TRIGGER TO SQLTUNING;  
GRANT CREATE ANY VIEW TO SQLTUNING;  
GRANT CREATE PROCEDURE TO SQLTUNING;  
GRANT CREATE SESSION TO SQLTUNING;  
GRANT CREATE TRIGGER TO SQLTUNING;  
GRANT CREATE VIEW TO SQLTUNING;  
GRANT DROP ANY OUTLINE TO SQLTUNING;  
GRANT DROP ANY PROCEDURE TO SQLTUNING;  
GRANT DROP ANY TRIGGER TO SQLTUNING;  
GRANT DROP ANY VIEW TO SQLTUNING;  
GRANT SELECT ON SYS.V_$SESSION TO SQLTUNING;  
GRANT SELECT ON SYS.V_$SESSTAT TO SQLTUNING;  
GRANT SELECT ON SYS.V_$SQL TO SQLTUNING;  
GRANT SELECT ON SYS.V_$STATNAME TO SQLTUNING;
```

DBMS Support and System Requirements



DBMS Support

- Oracle® 8i-11g
- Sybase® 12.5 - 15.0
- IBM® DB2® for LUW 8.0 - 9.0
- Microsoft® SQL Server 2000, 2005 and 2008

System Requirements

- Microsoft Windows 2003, XP, Vista (32 bit), Red Hat Enterprise Linux 5.0 (32 bit), or SUSE Linux Enterprise Server 10 (32 bit)
- Sun Java 2 Standard Edition 5.0 Update 11 for Microsoft Windows or Linux: Sun Java 2 Standard Edition 5.0 Update 11 for Linux x86
- 1024 MB memory
- 500 MB disk space

Agenda

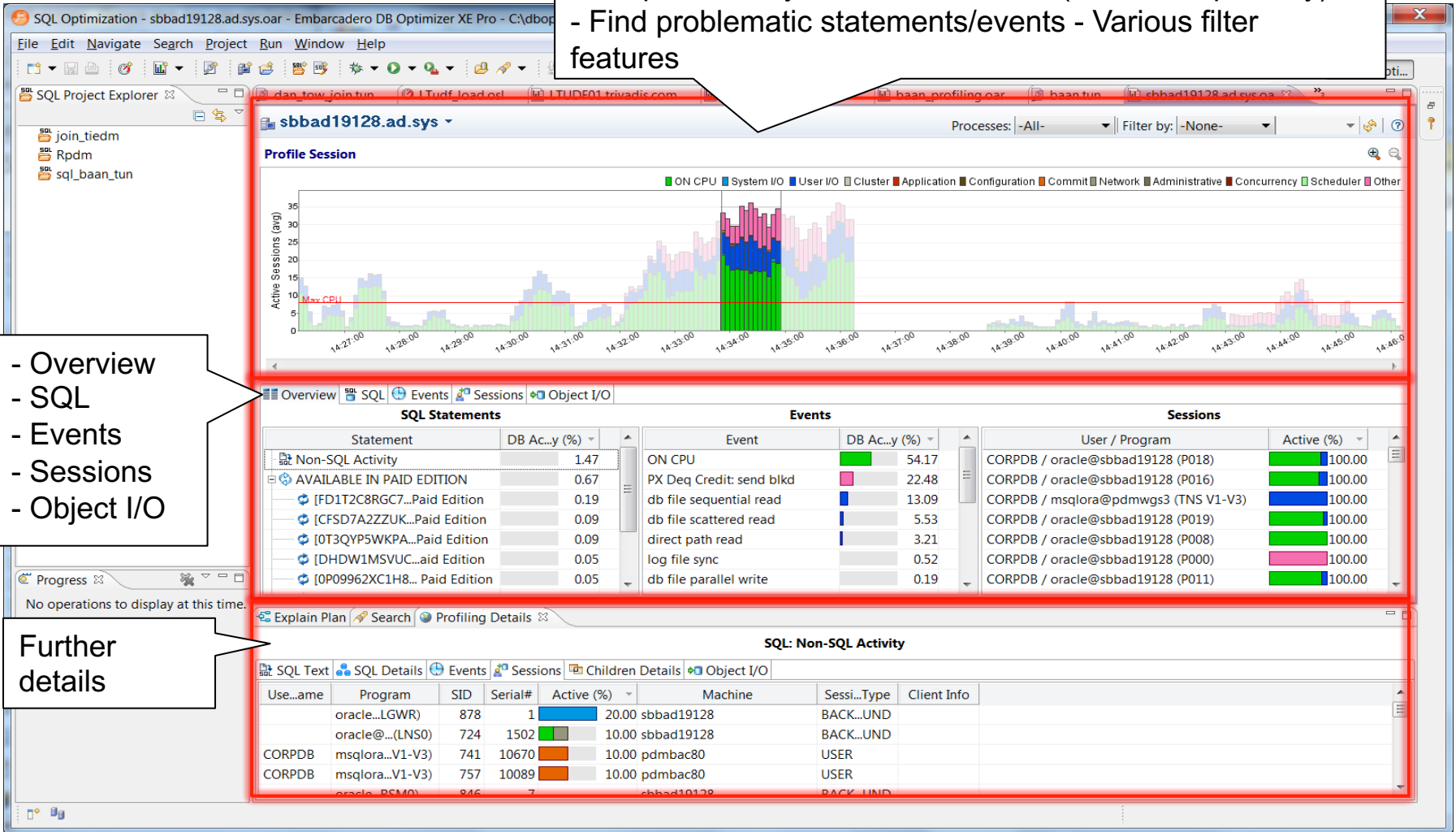


Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

Profiling One View with 3 Windows Sections

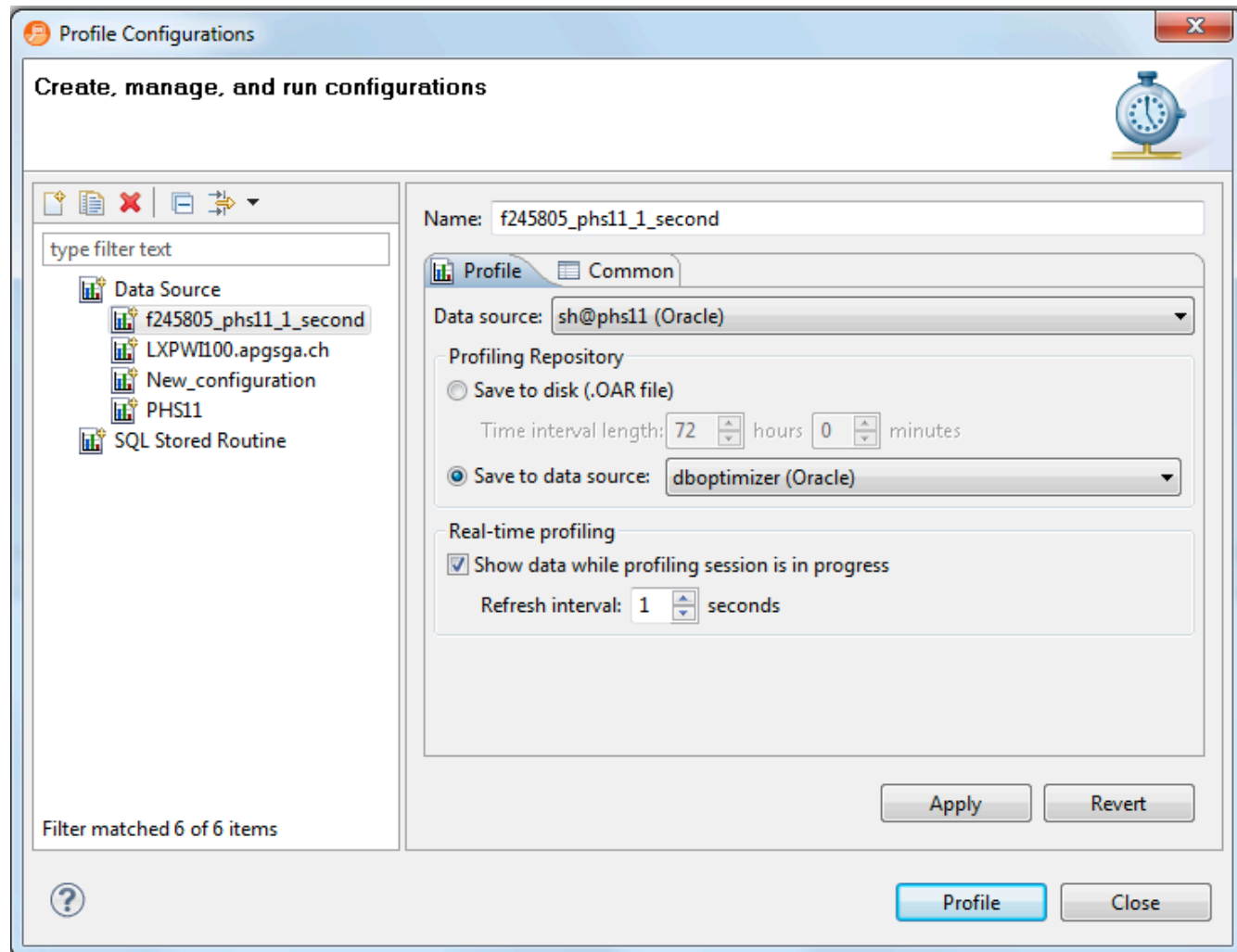
- Average Active Sessions
- Snapshot Analysis available offline (from file/repository)
- Find problematic statements/events - Various filter features



- Overview
- SQL
- Events
- Sessions
- Object I/O

Further details

Profile Configuration



Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

Identify Tuning Candidates



- Identify Tuning Candidates
 - Ad hoc SQL
 - Database Objects
 - SQL Files
 - Active SQL in SGA
- Option 1: Generate Alternative Statements (Brute Force Tuning Approach)
 - Find statements through hint injection
 - Find rewritten statements
 - Execute these statements (cancel execution after elapsed time of 150% of baseline)
 - Compare to baseline (original statement) to find "best" plan
 - Apply best plan using outline wizard
- Option 2: Detailed SQL Analysis
 - SQL Diagram
 - Index Analysis
 - Table Statistics
 - Column Statistics and Histograms
 - Outlines

Tuner Option 1 – After Generating Cases



SQL Optimization - join_tiedm/dan_tow_join.tun - Embarcadero DB Optimizer XE Pro - C:\dboptimizer\workspace

File Edit Navigate Search Project Run Window Help

SQL Project Explorer: join_tiedm, join_view, Rpdm, sql_baan_tun

Input Overview Analysis

Overview

Tuning Statements ☒ Generate cases ☒ Perform detail analysis ☒ Execute each generated case 2 times

Name	Schema	Text	Tables		Views		Time		Analysis	
			Elap... (s)	Improved (s)	Cases	Indexes				
SELECT 1	SYS	select from	8	0	6.79	6.79	12	1	9	6

Generated Cases

SQL Statements and Cases		Cost	Execution Statistics						
Name	Text	Value	Result	Elapsed Time (s)	Result	Rows ...urned	Physi...eads	Logi...eads	CPU Time
SELECT 1	select from udf.orders,	5060.0		6.79		0	0	0	
DYNAMIC_SAMPLING		3176.0		5.68		0	0	0	
ALL_ROWS				4.19		0	0	0	
FULL				4.42		0	0	0	
LEADING7				3.97		0	0	0	
LEADING2				3.68		0	0	0	
LEADING1		7884.0		4.01		0	0	0	
ORDERED		7890.0		5.56		0	0	0	

The DYNAMIC_SAMPLING hint instructs the optimizer how to control dynamic sampling to improve server performance by determining more accurate predicate selectivity and statistics for tables and indexes. The case generation for this hint accepts a configurable list of parameters (any combination of integer values from 0 to 10).

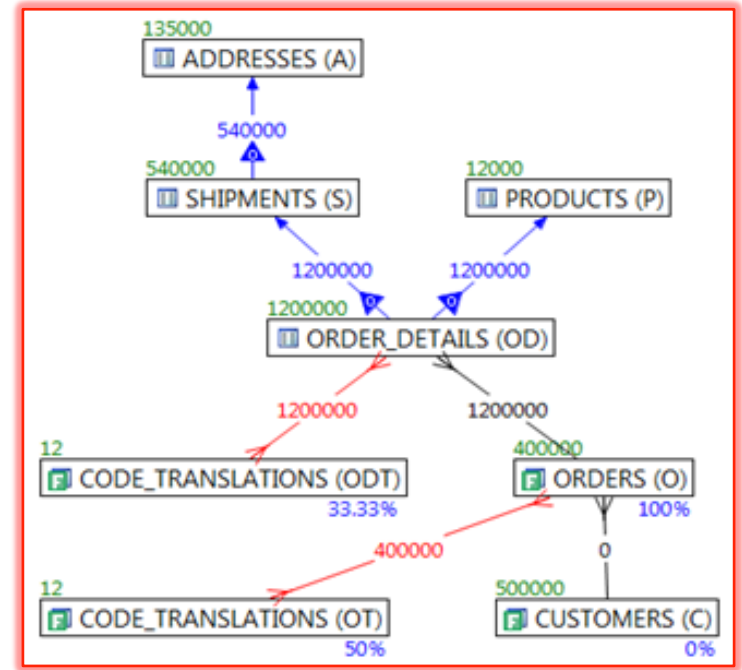
Progress: Show Count/Ratio Information, Show Count/Ratio Information, Untitled Tuning J...cuting cases..., Untitled Tuning ...cuting cases...

Explain Plan Search

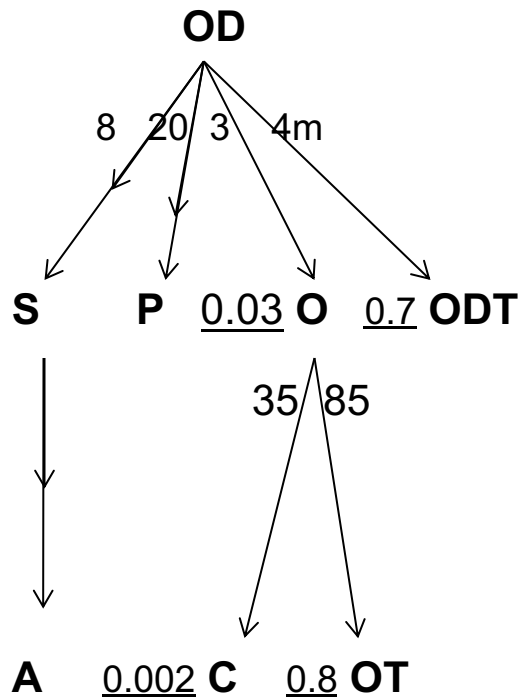
Show Count/Ratio Information

Option 2 – SQL Diagram

- ■ ■
- VST – Visual SQL Tuning using SQL Diagram to
 - Faster understanding of the query
 - Display table relationships and sizes graphically
 - Detecting bottlenecks in the query execution plan easier
 - Similarities to SQL Diagram from Dan Tow
- Calculates the join ratio between tables and filter ratios
- Usage of selected object in SQL diagram is highlighted in SQL statement



Diagramming SQL by Dan Tow-Method



Optimal Join Order

C,O,OT,OD,ODT,P,S,A

```
SELECT ....
```

```
FROM Orders O, Order_Details OD, Products P,
Customers C, Shipments S, Addresses A,
Code_Translations ODT, Code_Translations OT
WHERE C.Last_Name like :Last_Name||'\%'
```

```
AND C.First_Name like :First_Name||'\%'
```

```
AND OD.Order_ID = O.Order_ID
```

```
AND O.Customer_ID = C.Customer_ID
```

```
AND OD.Product_ID = P.Product_ID (+)
```

```
AND OD.Shipment_ID = S.Shipment_ID (+)
```

```
AND S.Address_ID = A.Address_ID (+)
```

```
AND O.Status_Code = OT.Code
```

```
AND OT.Code_Type = 'ORDER_STATUS'
```

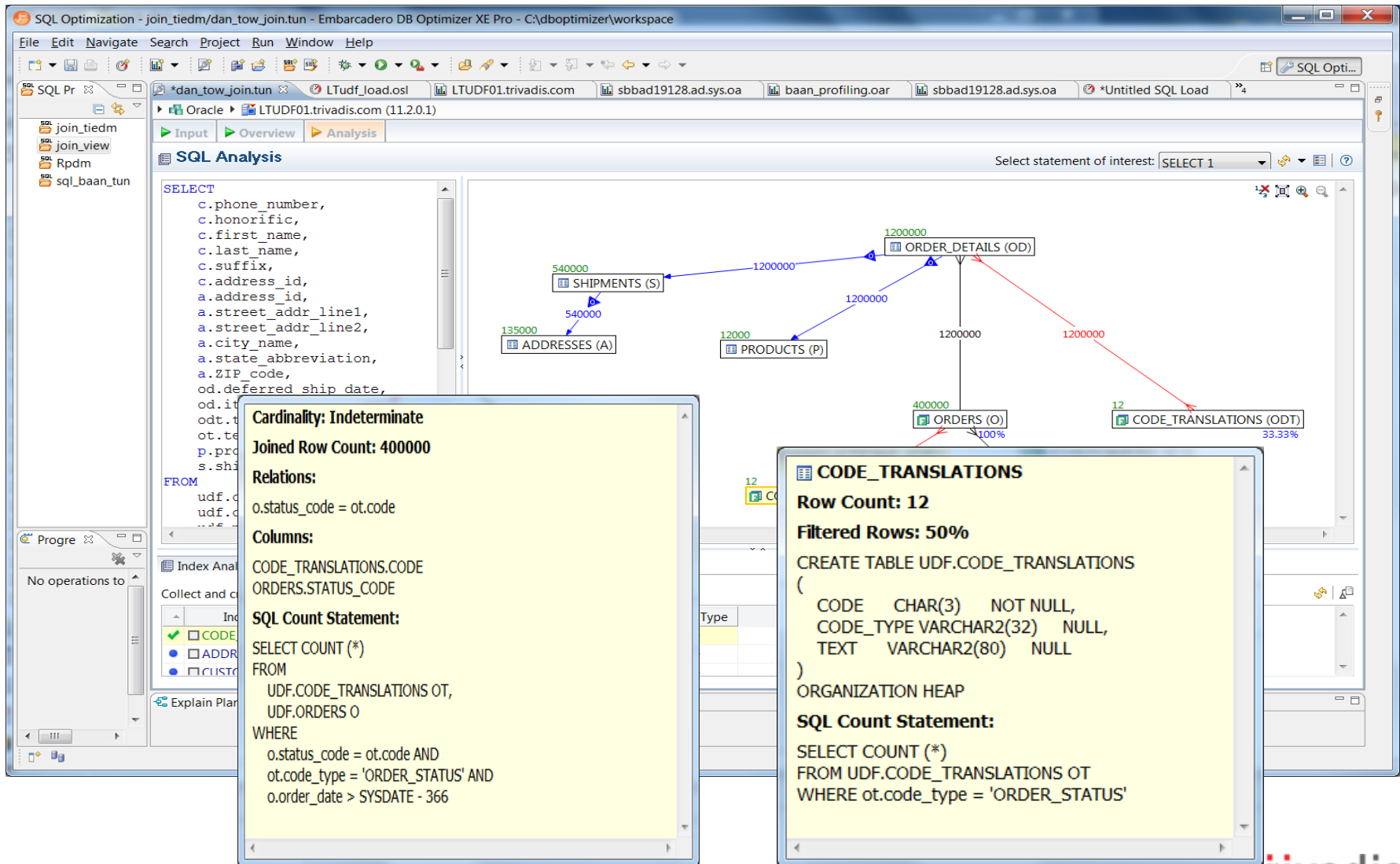
```
AND OD.Status_Code = ODT.Code
```

```
AND ODT.Code_Type = 'ORDER_DETAIL_STATUS'
```

```
AND O.Order_Date > :NOW -366
```

```
ORDER by C.Customer_ID, O.Order_ID Des;
```

SQL Diagram With Ratios



SQL Analysis



■ Index analysis:

- listing of used indexes,
- not used indexes,
- available indexes and
- suggested indexes through different colours

	Index Name	Table Owner	Table Name	Column Name	Index Type
✓	CODE_TR_ON_PKEY	UDF	CODE_ACTIONS	CODE_TYPE, CODE	Unique
•	ADDRESS_PKEY	UDF	ADDRESSES	ADDRESS_ID	Unique
•	CUSTOMER_PKEY	UDF	CUSTOMERS	CUSTOMER_ID	Unique

- Highlight SQL rewrite suggestions to make sure that the result is not affected , eg. missing join (red –table)
- Detail object statistics
- Generating Outlines

Join Between 2 Views

DEMO

The screenshot displays the Embarcadero DB Optimizer XE Pro interface. The main window shows a SQL Analysis of a query. The query text is:

```
SELECT *
FROM
  safe_doc_sp sp,
  safe_doc_rp rp
WHERE sp.r_object_id = rp.r_object_id;
```

The diagram on the right illustrates the join between two views: **SAFE_DOC_SP (SP)** and **SAFE_DOC_RP (RP)**. The join is represented by a red line with a double-headed arrow, indicating an equijoin.

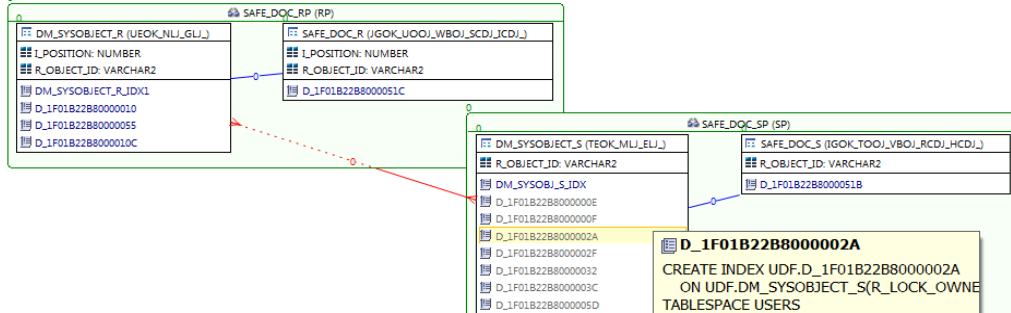
The bottom section shows the 'View table statistics' table, which lists statistics for various objects. The table has columns for Object, Table Owner, Table Name, Statistics Status, Days Since Taken, No of Rows, Blocks, Empty Blocks, Sample Size, Monitoring, No Inserts, and No Updates.




Object	Table Owner	Table Name	Statistics Status	Days Since Taken	No of Rows	Blocks	Empty Blocks	Sample Size	Monitoring	No Inserts	No Updates
UDF	DM_SYS...	ECT_S	Statistics OK	1	370176	25522	0	2000	YES		
UDF		SAFE_DOC_S	Statistics OK	1	189258	6670	0	2000	YES		
UDF	DM_SYS...	ECT_R	Statistics OK	1	1147717	7300	0	2000	YES		
UDF		SAFE_DOC_R	Statistics OK	1	423672	1882	0	2000	YES		

The bottom status bar shows the SQL Statement: `select from safe_doc_sp, safe_doc_rp`.

Select statement of interest:

```
SELECT *
FROM
    safe_doc_sp sp,
    safe_doc_rp rp
WHERE sp.r object id = rp.r
```



 Explain Plan   Search

select from safe_doc_sp, safe_doc_rp

Plan Cost				Estimated Statistics					A...
Operation	Cost	Opera...Cost	Result	Cardinality	Bytes	CPU Cost	IO Cost	Optimizer	Starts
SELECT STATEMENT	39198.0	0.0		231130	179588010	2708066154	39007	ALL_ROWS	
HASH JOIN	39198.0	13988.0		231130	179588010	2708066154	39007		
TABLE ACCESS - UDF.SAFE_DOC_S	1828.0	1828.0		189258	45421920	284072505	1808	ANALYZED	
HASH JOIN	23382.0	10137.0		451518	242465166	2150173277	23231		
HASH JOIN	6277.0	3732.0		450962	30665416	1123088367	6198		
TABLE ACCESS - UDF.SAFE_DOC_R	520.0	520.0		433673	11275498	121820800	511	ANALYZED	
TABLE ACCESS - UDF.DM_SYSOBJECT_R	2025.0	2025.0		1147717	48204114	660276522	1979	ANALYZED	
TABLE ACCESS - UDF.DM_SYSOBJECT_S	6968.0	6968.0		370176	173612544	762929712	6914	ANALYZED	

Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

Load Test

Execute Query using multiple sessions to simulate a concurrent user load
(typically suited for select statements only)

SQL Optimization - Untitled SQL Load - Embarcadero DB Optimizer XE Pro - C:\dboptimizer\workspace

File Edit Navigate Search Project Run Window Help

SQL Pr *dan_tow_join.tun LTudf_load.osl LTUDF01.trivadis.com sbbad19128.ad.sys.oa baan_profiling.oar sbbad19128.ad.sys.oa *Untitled SQL Load

LTUDF01.trivadis.com Change data source

Ad hoc SQL SQL file

```
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_name, sp.r_lock_date,sp.alsc_return_date
      rp.keywords,rp.i_folder_id,rp.a_effective_label
from udf.safe_doc_sp sp,
     udf.safe_doc_rp rp
where sp.r_object_id=rp.r_object_id
      and sp.a_retention_date > sysdate -30
      and sp.r_lock_owner like 'HUGO%'
      and rp.i_partition > 10
      and rp.a_effective_date > sysdate-30;
```

Number of parallel sessions: 2

Execution end condition:

Time: 0 hours 10 minutes

Number of executions: 3

Sleep between executions

Fixed delay: 250 ms

Random interval: 1250 ms 2251 ms

Progre

No operations to dis

type filter text

Operation	Plan Cost	Cost	Opera...Cost	Result	Cardinality	Bytes	CPU Cost	IO Cost	Optimizer	Starts
SELECT STATEMENT		10.0	0.0		1	222	80455	10	ALL_ROWS	
NESTED LOOPS		10.0	0.0		1	222	80455	10		
NESTED LOOPS		9.0	0.0		1	201	71433	9		

SQL Log

SQL Statement	Date	Host/Server	DBMS	Service	User	Time (ms)	Source
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:24:4...	LTUDF01.triv...	Oracle	UDFORA...	sys	2.73	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:24:4...	LTUDF01.triv...	Oracle	UDFORA...	sys	2.33	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:24:4...	LTUDF01.triv...	Oracle	UDFORA...	sys	2.64	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:24:4...	LTUDF01.triv...	Oracle	UDFORA...	sys	8.92	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:24:4...	LTUDF01.triv...	Oracle	UDFORA...	sys	1416.67	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:24:4...	LTUDF01.triv...	Oracle	UDFORA...	sys	924.49	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:23:0...	LTUDF01.triv...	Oracle	UDFORA...	sys	25.84	--

Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

SQL Editor



- Code assist (e.g. code completion)
- Code formatter
- Code corrections and transformations
(quick fixes, e.g. for Cartesian join)

```
SELECT ...  
FROM  
    udf.ttiedm110600 a,  
    udf.ttiedm100600 b  
WHERE  
    b.t$eitm (+) = a.t$eitm AND  
    b.t$revi (+) = a.t$revi AND  
    (b.t$exdt > ' 04/03/2011  
00:00:00') AND  
    a.t$cmtp = 2 AND  
    a.t$comp > 'TRVP065860000' AND  
    b.t$indt <= '04/03/2011  
23:59:00' AND  
    b.t$rele = 1
```

```
SELECT ...  
FROM  
    udf.ttiedm110600 a,  
    udf.ttiedm100600 b  
WHERE  
    b.t$eitm (+) = a.t$eitm AND  
    b.t$revi (+) = a.t$revi AND  
    (b.t$exdt (+) > ' 04/03/2011  
00:00:00') AND  
    a.t$cmtp = 2 AND  
    a.t$comp > 'TRVP065860000' AND  
    b.t$indt (+) <= '04/03/2011  
23:59:00' AND  
    b.t$rele (+) = 1
```

SQL IDE – Fix Missing Outer Join



The screenshot shows the SQL Optimization interface for an untitled tuning job. The 'Overview' tab is active, displaying a table of tuning statements and a table of generated cases. A tooltip is visible over the 'Invalid or missing outer join transformation' error in the 'Generated Cases' table.

Tuning Statements

Name	Schema	Text	Tables	Views	Elap... (s)	Impr... (s)	Cases	Indexes
SELECT 1	UDF	select from	2	0	1.52	1.52	7	0 5

Generated Cases

Name	Text	Value	Elapsed Time (s)	Physi...eads	Logi...eads	CPU Time (s)
INDEX_SS	Click here to view SQL text	8.0				
INDEX_FFS		8.0				
ALL_ROWS		122.0				
INDEX_COMBINE		1825.0				
ORDERED		2876.0				
USE_HASH		2986.0				
NO_USE_NL		3986.0				
Invalid or missing outer join transformation		4939.0	1.92	10450	31142	0.10

Fixed:
- invalid or missing outer join.
The result set may be altered.

SQL Log

SQL Statement	Date	Host/Server	DBMS	Service	User	Time (ms)	Source
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:28:3...	LTUDF01.triv...	Oracle	UDFORA...	sys	2.55	--
select sp.object_name,sp.title,sp.subject,sp.acl_name,sp.owner_nar	2011-03-25 12:28:3	LTUDF01 triv	Oracle	UDFORA	sys	19.61	--

Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

DB Optimizer XE for a DWH of a Financial Institute



- "Optimizing Monthly Processing" (>30h)
- DB Optimizer XE was used as a secondary information source only (Profiling Analysis)
- The primary information source was a set of log files to get a better understanding of the contribution of the components to the end-to-end runtime of the overall processes (Runtime Log Analysis)
- Among the first question was about the overhead/load DB Optimizer XE is producing (see next slide)

What is the Profiling Overhead of DB Optimizer XE?

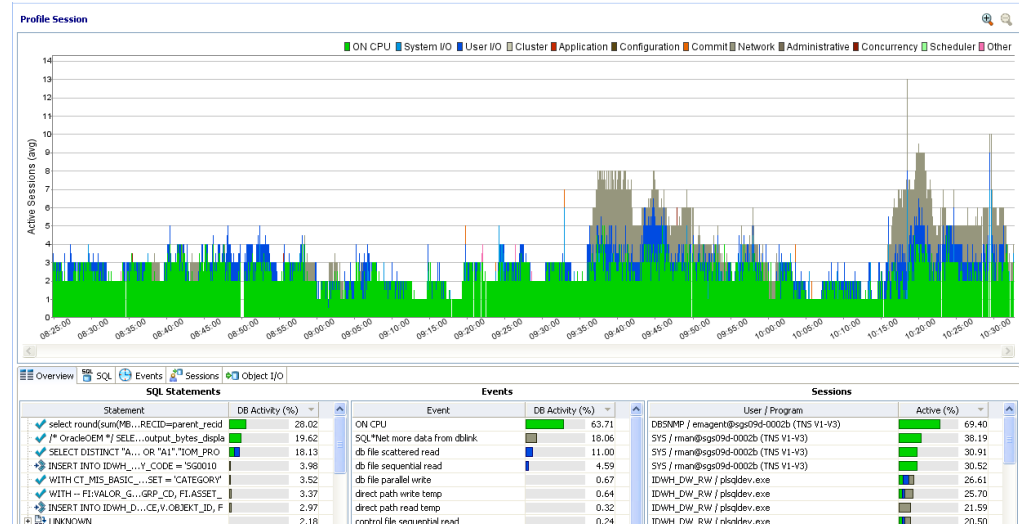


- Analysis of 10046 Trace file initiated by DB Optimizer XE profiler
 - 8 JDBC sessions created
 - A lot of dictionary queries are just executed once, generating insignificant load
 - Every second v\$session is queried (wait_time != 0 AND status = 'ACTIVE')
 - Based on v\$session query result additional information is retrieved periodically and via a single query for multiple objects using in-lists, e.g. V\$SQL, V\$SYSTEM_EVENT
- Overhead cannot be generally defined in a fixed percentage of the DB load, nonetheless, the load by DB Optimizer XE is considered to be very small and additional issues on the profiled database instance are not expected.

Profiling Analysis



- The graph presents CPU usage as green values, other colors are considered bottlenecks.
- The goal is to identify top resource consumers for optimization.
- The application context of the SQL statement (e.g. PL/SQL unit, PowerCenter workflow, load process, etc.) is not known by the profiler.
- It is quite difficult to predict the impact of the optimization on runtime of the overall process, especially if processes are executed in parallel and enough system resources are available.
- E.g. disabling the Oracle Enterprise Manager activities in the above chart, which are in fact the top CPU resource consumers will reduce the runtime of the overall monthly process by 0 seconds, since enough CPU resources are available (and no other resources are consumed).
- Nonetheless the profiling results combined with application knowledge will potentially identify SQL statements on the critical path, where an optimization of the SQL statement will also reduce the runtime of the overall process.



General Limitations on Data Accuracy



- DB Optimizer XE is sampling the Oracle V\$SESSION data (active sessions or sessions waiting for resources) every second.
- This means that short running SQL statements are usually missed a lot of times. However, short running, often executed statements should be recognized nonetheless, but the metric DB Activity (%) is typically reported too low. In these cases other V\$SQL based metrics will provide more accurate information.
- Generally speaking this tool is better suited to identify bottlenecks of long running statements than often executed short running statements.

Instrumentation Bug 5928612



- The profiled database instances run on an Oracle Database Server version which has a known instrumentation bug. Here's the description of Oracle bug number 5928612:
If a job is run through DBMS_JOB then the columns SQL_ADDRESS, SQL_ID and SQL_HASH_VALUE are not populated in V\$SESSION which can hinder performance monitoring. (the columns show as 0). A similar issue exists for DBMS_SCHEDULER under <Bug:5140631>
- The bug should be fixed in the current 11g and 10gR2 releases.
- DB Optimizer XE very much depends on correct V\$SESSION entries as other tools like Oracle Enterprise Manager do.
- Since the major part of the daily loads are executed via ad-hoc DBMS_JOB jobs this will certainly hinder performance monitoring. In fact the daily load currently cannot be profiled in a way to identify long-running statements using such a profiler.

Findings Concerning DB Optimizer XE



- One instance of DB Optimizer XE may profile just one DB instance in parallel
- Starting multiple DB Optimizer XE instances requires multiple Eclipse workspaces
- Profiling of 6 DB instances in parallel were possible (for a short time) – The longer a profiling session runs the more memory is needed.
- Saving files after an error was generally impossible (either an index out of bounds exception was raised or the save process was hanging)
- Errors occurred also without PC resource bottlenecks
- Long running profile results probably should be stored on a unrelated DB instance instead of local files (or the enterprise features should be considered) – to be verified
- Tuner could not be used because of restricted access rights, long parsing times, parsing errors

Findings Based on Profiling Analysis



- Long-running statements (0.5h) joining several single row context tables
 - Confirms finding based on Runtime Log Analysis
 - Huge performance improvement through use of SYS_CONTEXT proved by a test-case.
 - Expected performance improvement from 3h to less than 1h
- Long-running statement (1.2h) caused by an suboptimal plan
 - Oracle did not find a Hash join based plan because of various IN-Lists combinations
 - Rewrite of statement possible with application knowledge to allow the optimizer to find a hash join plan leads to major runtime improvement (less than 5 minutes)
- Periodical executed, I/O resource intensive status queries identified
- ALTER TABLESPACE ADD DATAFILE statements found during month end processing

Agenda



Data are always
part of the game.

- Introduction
- Installation
- Profiling
- (Visual) SQL Tuning
- Load Testing
- SQL IDE
- First Real World Experiences
- Conclusion

Assessment



■ Pros

- Independent of Oracle version, edition and licensed options
- Easy to install
- Self-explanatory interface
- Profiler allows quick access to top resource consumers and related statements
- Good visualization of complex SQL statements (VST) incl. view expansion
- Faster SQL Tuning process regarding
 - Index analysis
 - Statistics validation
 - Load test and generating cases
- Explain plan supports binds (for standard types only) and parsing user

■ Cons

- Unsuccessful parse makes Tuner and SQL Editor unusable
- Parser is sometimes really slow
- Occasionally an inconsistent internal status avoids save to file (index out of bounds error, crashes, spins)
 - Profiler
 - Tuner
- Plan visualization in other IDEs are better (e.g. execution sequence in SQL Navigator, PL/SQL Developer)
- Named user licensing model is locked to a single or few client machines

Conclusion



- The Profiler and the VST alone makes DB Optimizer XE a valuable tool for every Oracle DB related, performance optimizing job
- The parsing and saving errors are considered bugs and we expect that they should be addressed in coming releases
- We recommend to start licensing negotiations with Embarcadero Technologies to enable every Trivadis Performance Consultant to use the tool as part of the APM survival kit
- Additional Technology projects for SQL Server is recommended too

■ ■ ■ Thank you!



www.trivadis.com

trivadis
makes IT easier. ■ ■ ■



Basel

Bern

Lausanne

Zurich

Düsseldorf

Frankfurt/M.

Freiburg i. Br.

Hamburg

Munich

Stuttgart

Vienna